Assignment 8-5 Integer Subtraction 4

You are given two positive integers a and b. The problem is to subtract b from a. ($b \leq a < 10^{250}$)

Input

The input consists of t ($30 \leq t \leq 40$) test cases. The first line of the input contains only positive integer t. Then t test cases follow. Each test case consists of two lines which give the two positive integers a and b ($b \leq a < 10^{250}$), respectively.

Output

For each test case, you are to output a single line containing the difference $a - b$.

Sample Input

```
2
12345
12311
12345
12345
```

Sample Output

```
34
0
```

Part of the program

You are required to write the function `subtraction` to complete the following program which solves this problem. In your program, you cannot declare global variables or static arrays.

```
#include <iostream>
#include <cstring>
using namespace::std;

class HugeInt
{
public:
   int size;
   int *digit;

   // returns true if and only if hugeInt1 == hugeInt2
   bool equal( HugeInt hugeInt2 );

   // difference = minuend - subtrahend provided that minuend >= subtrahend
   void subtraction( HugeInt subtrahend, HugeInt &difference );
};
```

```
int main()
{
   char strA[ 251 ], strB[ 251 ];

   int T;
   cin >> T;
   for( int t = 0; t < T; ++t )
   {
      cin >> strA >> strB;

      HugeInt minuend;
      minuend.size = strlen( strA );
      minuend.digit = new int[ minuend.size ];
      for( int i = 0; i < minuend.size; ++i )
         minuend.digit[ i ] = strA[ minuend.size - 1 - i ] - '0';

      HugeInt subtrahend;
      subtrahend.size = strlen( strB );
      subtrahend.digit = new int[ subtrahend.size ];
      for( int i = 0; i < subtrahend.size; ++i )
         subtrahend.digit[ i ] = strB[ subtrahend.size - 1 - i ] - '0';

      HugeInt difference;
      minuend.subtraction( subtrahend, difference );

      for( int i = difference.size - 1; i >= 0; i-- )
         cout << difference.digit[ i ];
      cout << endl;

      delete[] minuend.digit;
      delete[] subtrahend.digit;
      delete[] difference.digit;
   }
}

// returns true if and only if hugeInt1 == hugeInt2
bool HugeInt::equal( HugeInt hugeInt2 )
{
   if( size != hugeInt2.size )
      return false;

   for( int i = 0; i < size; i++ )
      if( digit[ i ] != hugeInt2.digit[ i ] )
         return false;

   return true;
}

// difference = minuend - subtrahend provided that minuend >= subtrahend
void HugeInt::subtraction( HugeInt subtrahend, HugeInt &difference )
{
   if( equal( subtrahend ) )
   {
      difference.size = 1;
      difference.digit = new int[ 1 ];
      difference.digit[ 0 ] = 0;
      return;
   }






}
```